UNIVERSITY FOR DEVELOPMENT STUDIES, TAMALE

ASSESSMENT OF SOME ACCELERATION SCHEMES IN THE SOLUTION OF SYSTEMS OF LINEAR EQUATIONS.

SEIDU AZIZU

2015

UNIVERSITY FOR DEVELOPMENT STUDIES, TAMALE

ASSESSMENT OF SOME ACCELERATION SCHEMES IN THE SOLUTION OF SYSTEMS OF LINEAR EQUATIONS.

BY

SEUDU AZIZU (B.Ed. Mathematics)

(UDS/MM/0016/12)

THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, FACULTY OF MATHEMATICAL SCIENCES, UNIVERSITY FOR DEVELOPMENT STUDIES, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE DEGREE IN MATHEMATICS

JANUARY, 2015

DECLARATION

Student:

I hereby declare that this dissertation/thesis is the result of my own original work and that no part of it has been presented for another degree in this University or elsewhere:

Candidate's Signature:……………...……………. Date:……………………..………

Name: …………………………………………………………………...………

Supervisor:

I hereby declare that the preparation and presentation of the dissertation/thesis was supervised in accordance with the guidelines on supervision of dissertation/thesis laid down by the University for Development Studies.

Principal Supervisor's Signature:……………………….. Date:…………………

Name: ………………………………...………………………………………..…..

ABSTRACT

An assessment of acceleration schemes in the solution of systems of linear equations has been studied. The iterative methods:Jacobi, Gauss-Seidel and SOR methods were incorporated into the acceleration scheme (Chebyshev extrapolation, Residual smoothing, Accelerated gradient and Richardson Extrapolation)to speed up their convergence. The Conjugate gradient methods of GMRES, BICGSTAB and QMR were also assessed. The research focused on Banded systems, Tridiagonal systems and Dense Symmetric positive definite systems of linear equations for numerical experiments. The experiments were based on the following performance criteria: convergence, number of iterations, speed of convergence and relative residual of each method. Matlab version 7.0.1 was used for the computation of the resulting algorithms. Assessment of the numerical results showed that the accelerated schemes improved the performance of Jacobi, Gauss-Seidel and SOR methods. The Chebyshev and Richardson acceleration methods converged faster than the conjugate gradient methods of GMRES, MINRES, QMR and BICGSTAB in general.

ACKNOWLEDGEMENTS

First of all I express my gratitude to Almighty Allah for enabling me to complete this work successfully. Words of gratitude and appreciation do not always convey the depth of one's feelings, yet I wish to record my thanks to my most respected Supervisor Dr. Stephen B. Twum, for his wisdom, kindness and constructive criticisms during the research work.

I further wish to acknowledge Mr. Mohammed Muniru Iddrisu for his invaluable, intellectual suggestions that he rendered to me during the course of the research work.

I am grateful to the Head of Mathematics Department,Dr. Yakubu Ibrahim Seini for his frequent motivational advice towards research at the lecture hall.

I wish to express my heartfelt thanks and deep gratitude to my wife, Baliya Kassim for her prayers and encouragement during my studies. I am also thankful to the rest of my family and friends for their supports and frequent prayers.

This acknowledgment would remain incomplete without expressing my heartfelt gratitude to my course mates: Imoro Rafiatu, Etwire Christian John and Emmanuel Maurice Arthur for their support and frequent suggestions.

DEDICATION

I dedicate this work to my parents Mallam Seidu Sulaiman Imam andMagaajiya Zuwaira Yussif for their support and foresight towards my education.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

AGS	:	Accelerated Gradient Scheme
BIGGSTAB	:	Biconjugate Gradient Stabilized
CES	:	Chebyshev Extrapolation Scheme
CG	:	Conjugate Gradient
Eqn	:	Equation
GMRES	:	Generalized Minimal Residual
MAXIT	:	Maximum Number of Iteration
MINRES	:	Minimal Residual
QMR	:	Quasi-Minimal Residual
RS	:	Residual Smoothing
RES	:	Richardson Extrapolation Scheme
RHS	:	Right- Hand Side
SOR	:	Successive Over Relaxation
SPD	:	Symmetric Positive Definite
SLE	:	Systems of Linear Equations

CHAPTER ONE

INTRODUCTION

1.0Introduction

Many practical problems can be expressed in the form of systems of linear equations $Ax=b$, whereA is a known matrix, b and $x$are respectively known and unknownvectors. System of linear equations play importantrole in Economics, Engineering, Physics, Chemistry, Computer Science and other fields of Pure and Applied Sciences (Iqbal*et al*.2012).

Convergent numerical sequences occur quite often in natural Science and Engineering.Some of such sequences converge very slowly and their limits are not available without suitable convergent acceleration methods. Santos and Linhares (1986) stated that there are two classes of iterative methods: linear stationary and linear nonstationary. The choice of a method for solving linear systems will often depend on the structure of the matrixA. Direct methods are mostly chosen when the size of the system is small and require solving with several right hand side vectors b while maintaining the same coefficient matrix A. However, they have some pitfalls such as division by zero, round-off errors and huge storage requirement. Iterative methods are chosen for large and sparse systems. An important feature of the iterative methods is that they are relatively faster than direct methods and occupy very little space and work indirectly with the original matrixA.The convergences of these stationary iterative methods are known to be relatively slow and need to be accelerated. Therefore, in this research we access the performance of some known acceleration schemes at improving the convergence of some selected stationary iterative methods.

1.1Background of the Study

One of the fundamental building blocks of numerical computing is the ability to solve linear system*s* .Systems of linear equations may consist of two or more linear equations. A solution exists for it if and only if the solution satisfies all equations of the system. A System of linear equations has the general form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

Wherex_j(j= 1, 2, . . ., n) are unknown variables, a_{ij}(i, j= 1, 2, . . . ,n) are the coefficients, and

b_i(i= 1, 2, . . .,n) are right hand side (RHS) constants. The first subscript identifies the row of the equation and the second subscript identifies the column of the system of equations. A system of linear equations can also be written as a matrix equation $Ax= b$, where:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{11} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{22} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

The term "iterative method" refers to a wide range of techniques that use successive approximations to obtain more accurate solutions (Gismalla, 2014). The two classes of iterative methods are Stationary and Nonstationary. Stationary methods are the classicaliterative methods such as Jacobi, Gauss-Seidel and Successive Over relaxation (SOR). The Nonstationary methods are more recent and are renamed the Krylov subspace methods (Saad, 2003). These are the Conjugate Gradient (CG), Biconjugate Stabilized (BICGSTAB), General Minimal Residual (GMRES), Minimal Residual (MINRES) and Quasi Minimal Residual (QMR). The Nonstationary methods are based on the idea of sequences of orthogonal vectors (Gutknecht and Rollin, 2002).

The classical iterative methods are incorporated into acceleration schemes to speed up convergence. Examples of acceleration schemes are Chebyshev Extrapolation scheme, Richardson extrapolation, Accelerated Gradient and Residual Smoothing. These schemes will be the focus of attention and discussion in this study.

1.2Statement of the Problem

Systems of linear equations arise from almost every field of mathematical applications, so the problem of solving linear system is of great importance. Numerous methods (Direct and Iterative) have been in existence for solving linear system. The direct methods find an exact solution and suitable for small system. However, they are subjected to round-off errors and are computational expensive. The Iterative methods (stationary and nonstationary) are suitable for large and sparse system but not suitable for particular matrix structure. The convergence of stationary iterative methods is relatively unsatisfactory (slow convergence).

However, many matrix problem are structured (Tridiagonal, Banded, Symmetric Positive Definite etc)where fast convergence methods are required to solve such systems. The Jacobi and Gauss-Seidel methods may or may not converge when the coefficient matrix A is not diagonally dominant and when the spectral radius of these methods is greater than 1(Burden and Faires, 2011). The successive over-relaxation method improves the convergence of Gauss-Seidel method. But fails to converge if the relaxation parameter is outside the range (0 2).The stationary iterative methods need to be accelerated using effective acceleration schemes. The problem is to assess which of the iterative methods are improved by which acceleration scheme. Therefore the acceleration schemes: Chebyshev extrapolation, Residual smoothing, Richardson extrapolation and Accelerated gradient would be applied to assess the convergence of the named iterative methods.

1.3 Objectives

The objective of the study is to assess the performance ofChebyshev Extrapolation Scheme, Accelerated Gradient Scheme, Residual Smoothing Scheme and Richardson Extrapolation scheme for solving systems oflinear equations.

1.3.1Specific Objectives

The specific objective are to:

1. incorporate stationary iterative methods into to the identified acceleration schemes
2. determine which iterative methods are improved by which acceleration schemes.
3. solve problem involving Tridiagonal, Banded, Dense and Symmetric Positive Definite systems of linear equations of different sizes.
4. compare the performance of the acceleration schemes with the known Krylov subspace methods.

1.4 Research Questions

1. Which schemes do better than the others for a given iterative method?
2. What characteristics of the systems influence performance of the acceleration scheme?
3. Which schemes improve the convergence of the classical iterative methods for the entire chosen systems of linear equations?

1.5Significance of the Study

1. Assessment of the relative efficiencies of the acceleration schemes would provide insight towards more economical and faster ways to solve linear systems of equations arising in applications.
2. The study is expected to provide further information about convergence and relative residual for each identified linear systems.
3. The result should be of interest to the wider research community in the field of Engineering, Science, Computing and Business.

1.6 Conclusion

This chapter explained the introduction, background of the study, statement of problem, the problem, research questions and the significance of the study. It also provided the basic concepts of systems of linear equations.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This chapter reviews the literature on the relevant concepts relating to this topic. These are specifically: What is a Systems of Linear Equations (SLE), Sources to SLE, Some special matrices and their properties, Stationaryiterative methods, Convergence of stationary iterative methods, Spectral Radius, Acceleration Schemes and related works.

2.1Systems of Linear Equations

A system of linear equation in n variables is a set of n equations with each equation being linear in n variables. It is generally denoted by the equation $Ax = b$,where A is a square matrix, x unknown vector and b a known vector. A solution is a set of numerical values $x_1, x_{2,} x_3, \ldots.. x_n$that satisfies all the equations (Fink and Mathews,2004).Various methods have been introduced to solve systems of linear equations. There is no single method that is best for all situations. These methods should be determined according to speed and accuracy. The methods for systems of linear equations can be divided into two groups: Direct and Iterative Methods. The approximate methods that provide solutions for systems of linear equations are called iterative methods. They start from an initial guess and improve the approximation until an absolute error is less than the pre-defined tolerance.

Iterative methods are suitable for very large systems oflinear equations. These methods are very effective concerning computer storage and time requirements. One of the advantages of using iterative methods is that they require fewer multiplications for large systems. They are fast and simple to use when coefficient matrix is sparse. Advantageously they have fewer rounds off errors as compared to other direct methods.

On the other hand, the direct methods, aim to calculate an exact solution in a finite number of operations.

The solution of the system is very important in scientific computing. If the linear system Ax = b has a unique solution x, for every right-hand side b then, the system is said to be nonsingular. Usually, one refers to the associated matrix A rather than the system of equation itself as being nonsingular, since the property is determined by the coefficients matrix A and does not depend on the right-hand side vector b.

A system of linear equation with coefficient matrix A and augmented matrix $\widetilde{A}$ has a solution if and only if rank A= rank $\breve{A}$. Again, a system with coefficient matrix A has a solution for any choice of $b_1 \dots \dots \dots .. b_m$ if and only if rank(A)is equal to the number of rows of A.

2.2Sources of Systems of Linear Equations

Sparse system occurred in a wide range of applications: Electrical power systems, linear and non-linear optimization and cryptography. Kelley (1995) is of the view that Information Retrieval Systems is one of the areas in which systems of linear equations mostly occur. In modern society, huge amounts of data are collected and stored in computers so that useful information can later be extracted. The growth of the internet has created the necessity for an efficient way to sort through massive amounts of documents for desired information. There have been multiple methods of information retrieval used to solve this dilemma, but one of the most basic forms uses matrices to model information and provide practical and efficient way to sort through it. A term document or text matrix is used to represent the frequency with which certain terms appear in a collection of documents, or a database. Each column of the matrix will represent a different document, and each row will represent a different term. In other words, the documents will make up the column space of the database.

A truss system is another situation that leads to systems of linear equations. Truss is an assemblage of long, slender structural elements that are connected at their ends. Trusses find substantial use in modern construction, for instance in towers and bridges. Trusses are often used to stiffen structures and most people are familiar with the often very elaborate systems of cross-bracing used in bridges. Trusses (structures made of beams) are light but rigid. They can support and divert very heavy loads, relative to their own weight. Trusses are primarily used in building construction and civil engineering projects.

Network systems in electrical engineering also lead to systems of linear equations. Today more than ever, electronics are an integral part of our everyday lives. They contribute to every aspect of our life from lighting the space around our work environments, to exploring uncharted territories. Behind each and every electrical appliance lies a vast system of electrical components that must function as a whole. Each component (resistors, capacitors, inductors, etc.) has specifications of their own, as does the final product that they are a part of, so engineers must design their devices to meet not only their intended purpose, but also the individual components. Vital to this is the analysis of currents and voltages throughout the electrical circuit. In a network model, you assume that the total flow into a junction is equal to the total flow out of the junction. Each junction in a network gives rise to a linear equation. You can analyze the flow through a network composed of several junctions by solving a system of linear equations.

2.3Some Special Matrices and their Properties

A matrix A is symmetric (Burden and Faires, 2011) if A= A^T and symmetric positive definite if $x^T \mathrm{A} x > 0$.When matrix A is positive definite the matrix that scales A is unique.Tridiagonal matrices crop up naturally in most areas of applied linear algebra.

Engineering problems often lead to coefficient matrices that are sparsely populated, meaning that most elements of the matrix are zero. If all the nonzero terms are clustered about the leading diagonal, then the matrix is said to be banded. All the elements lying outside the band are zero. The banded structure of a coefficient matrix can be exploited to save storage and computation time. If the coefficient matrix is also symmetric, further economies are possible. Most often we encounter pent diagonal (bandwidth = 5) coefficient matrices in the solution of fourth-order,ordinarydifferentialequationsbyfinite differences.Ofthese matrices are symmetric in nature.

In many applications, extremely large linearsystems that have a bandedstructure are encountered. Banded matrices often occur in solving ordinary and partial differential equations. It is advantageous to solve such linear systems, since they reduce the amount of storage used (Kincaid and Chenney,2008) .

2.4Structure of Banded and Tridiagonal Systems

Sparse matrices have the majority of their elements equal to zero. More significantly, a matrix is considered sparse if a computation involving it can utilize the number and location of its nonzero elements to reduce the run time over the same computation on a dense matrix of the same size.The finite difference and finite element methods for solving partial differential equations arising from mathematical models of continuous domain require sparse matrix algorithms.

In many situations for instance, boundary value problem for ordinary and partial differential equations matrix arise where large proportion of the elements are equal to zero. If the nonzero elements are concentrated around the main diagonal, the matrix is called band matrix. More precisely, a matrix A is said to be band if there are natural numbers p and q such that $a_{ij=0}$ if j$-i > p$ or $i - j > q$. The bandwidth (BW) = p $+q +$ 1. The band width is the maximum number of nonzero elements in any row. If p=q=1, then such band matrix is called tridiagonal as shown below.

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

Banded systems have the following advantages

1. When storing we do not store the elements outside the band.
2. One can take advantage of the band structure to reduce the number of operations.

2.5Stationary Iterative Methods

Iterative methods start from an initial guess $x_{0,}$which is successively improved until a sufficiently accurate solution is obtained. An important feature about these iterative methods is that they work indirectly with the original matrix A and only need extra storage for a few vectors. Iterative methods are more suitable for large and sparse matrix problems although they can also be used to deal with dense ones as well (Santos and Linhares, 1986).

The Gauss-Seidel method is like the Jacobi method, except that it uses updated values as soon as they are applied (Gismalla, 2014).Iterative methods are more suitable than direct methods for large linear systems(Yin, 2005).The generalization of these methods are proposed and their convergence properties are studied (Salkuyeh, 2007).

Iterative methods for solving general, large sparse linear systems have gainedpopularity in many areas of scientific computing.However, a number of efficient iterative solvers (LINPACK) were discovered and the increased need for solving very large linear systems triggered a noticeable and rapid shift toward iterative techniques in many applications. This trend can be traced back to the 1960s and 1970s when two important developments revolutionized solution methods for large linear systems and the realization that one can take advantage of “sparsity” to design special direct methods that can be quite economical.

2.5.1The Jacobi Method

The Jacobi method is an iterative algorithm used to solve linear equations. The solution of vector x is sought where $Ax = b$. Let $A = D + (L + U)$, where D, L and U represent the diagonal, lower triangular and upper triangular part of the coefficient matrix A respectively. Then the equation to be solved can be rewritten as:

$$Dx + (L + U)x = b \tag{2.1}$$

$$x = D^{-1}\left[b - (L+U)x\right] \tag{2.2}$$

If $a_{ii} \neq 0$ for each i, the definition for Jacobi method can be expressed as

$$x^{(k+1)} = D^{-1}\left[b - (L+U)x^{(k)}\right] \tag{2.3}$$

where k is the iteration count leading to the form:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \ldots n. \tag{2.4}$$

2.5.2The Steps for Jacobi Algorithm

1. Enter matrix A and choose an initial guess x^0, maximum number of iteration and tolerance (ε)
2. Find D, L and U of matrix A.
3. Set T = $-D^{-1}(L+U)$
4. Set C = $D^{-1}b$
5. Compute $x^{(k+1)} = Tx^{(k)} + C$
6. Stop if $\left\|x^{(k+1)} - x^{(k)}\right\|_\infty < \varepsilon$

2.5.3Convergence of Jacobi

According to Burden and Faires, (2011) the necessary and sufficient condition for convergence of the Jacobi methodis that matrix A is strictly diagonally dominant, then for any choice of $x^{(0)}$, both the Jacobi methods give sequences $\{x^k\}_{k=0}^{\infty}$ that converge to approximate solution SLE.

2.5.4 The Gauss-Seidel Method

Gauss-Seidel method is very similar to Jacobi method except that it updates the next iteration.

$$x^{(k+1)} = (D+L)^{-1}(-Ux^{(k)}+b) \tag{2.5}$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right) \tag{2.6}$$

2.5.5 The Steps forGauss-SeidelAlgorithm

1. Enter matrix A and choose an initial guess x^0, maximum number of iteration and tolerance (ε)
2. Find D, L and U of matrix A.
3. Find the inverse of (D + L).
4. Set T = $-(\mathrm{D}+\mathrm{L})^{-1}\mathrm{U}$
5. Set C = $(D+L)^{-1}$b
6. Compute $x^{(k+1)} = Tx^{(k)} + C$
7. Stop if $\left\|x^{(k+1)} - x^{(k)}\right\|_{\infty} < \varepsilon$

2.5.6 Convergence of Gauss-Seidel

The method is an improved version of the Jacobi method. It is defined on matrices with non-zero diagonals, but convergence is only guaranteed if the matrix is either diagonally dominant, or symmetric and (semi) positive definite. If a linear system has strictly dominant coefficient matrix and each equation is solved for each strictly dominant variable then, the Gauss-Seidel iteration will converge to x for any choice of x_0 (Jamil, 2012).

2.5.7The Successive Over Relaxation Method

Successive over-relaxation (SOR) is a numerical method originally used to speed up the convergence of the Gauss-Seidel method. The SOR method is devised by applying an extrapolation parameterω to the Gauss-Seidel method.This extrapolation parameter takes the form of a weighted average between the previous iteration and the computed Gauss-Seidel iteration successively (Gismalla, 2014).Thus the general iteration can be expressed as:

$$X^{(k+1)} = (1-\omega)X^{k} + \omega X^{(k+1)} \tag{2.7}$$

where the right-hand side $X^{(k+1)}$ should be calculated instead of the normal iteration expressions.

For Gauss-Seidel method, the SOR expression is

$$X^{(k+1)} = (1-\omega)X^{k} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right) \tag{2.8}$$

2.5.8 The Steps forThe Successive Over relaxation Algorithm

1. Enter matrix A and choose an initial guess x^0, maximum number of iteration, tolerance (ε) and $\omega \, \epsilon (0,2)$
2. Find D, L and U of matrix A.

3. Find the inverse of (D +ω L).
4. Set T = $-(\mathrm{D}+\omega\mathrm{L})^{-1}\mathrm{U}[\omega\mathrm{U}+(1-\omega)\mathrm{D}]$
5. Set C = $\omega(D+\omega L)^{-1}$b
6. Compute $x^{(k+1)} = Tx^{(k)} + C$
7. Stop if $\left\|x^{(k+1)} - x^{(k)}\right\|_{\infty} < \varepsilon$

2.5.9Convergence of Successive over relaxation

If the relaxation parameterω= 1, the SOR method is equal to the Gauss-Seidel method .The SOR method fails to converge if ω is outside the interval (0, 2). In general, it is not possible to compute in advance the value of ω that will maximize the rate of convergence of SOR.

2.6Nonstationary Iterative Methods

Varga (1962) reported that the nonstationary iterative methods werestandard iterative method for solving sparse linear systems. The conjugate gradient(CG) methodisone of the most important acceleration methods in solving large and sparse symmetric. The CG is Krylov subspace method based on Lanczos algorithm and is used to solve SPD linear system (Tullius, 2011).

The CG algorithm, applied to the system Ax = b, starts with an initial guess of the solution x^0 with an initial residual r_o and with an initial search direction that is equal to the initial residual: p_0= r_0. The idea behind the conjugate gradient method is that the residual r_k = b $-$A$x_k$is orthogonal to the Krylov subspace generated by b, and therefore each residual is perpendicular to all the previous residuals. The residual is computed at each step. The solution at the next step is found using a search direction that is only a linear combination of the previous search directions, which for $x^1$is just a combination between the previous and the current residual. Then, the solution at step k, x^k, is just the previous iterate plus a constant times the last search direction. Consider the problem of finding the vector x that minimizes the scalar function f(x)=$\frac{1}{2}x^T Ax - b^T x$ where the matrix A is symmetric and positive definite. Again f(x)is minimize when its gradient $\nabla f = Ax - b =$ 0. It implies that the minimization is equivalent to $Ax =$ b. Gradient methods accomplish the minimization by iteration starting with an initial vector x_0. Each iteration cycle k computes a refined solution

$$x^{(k+1)} = x^k + \alpha_k S_k \tag{2.9}$$

The step length α_k is chosen so that $x^{(k+1)}$ minimizes f($x^{(k+1)}$) in the search direction S_k.Again, $x^{(k+1)}$ must satisfy $Ax = b$ such that

$$A(x^k + \alpha_k S_k) = b \tag{2.10}$$

Introducing the residual $r_k = b - Ax^k$in equation (2.36) it becomes

$$\alpha_k AS_k = r_k \tag{2.11}$$

The immediate benefit of the search directions is that there is no need to store the previous search directions and the search is linearly independent of the previous directions. Again, at the solution of the next step, a new search direction is computed, as well as a new residual and new constant.

2.6.1 Generalized Minimal Residual (GMRES)

The Generalized Minimal Residual (GMRES)method computes a sequence of orthogonal vectors and combines these through least-squares method (Kelly, 1995). This method combines with preconditioning as discussed earlier in order to speed up convergence.

However, CG requires storing the whole sequence, so that a large amount of storage is needed. For this reason, restarted versions of this method are used. In restarted versions, computation and storage costs are limited by specifying a fixed number of vectors to be generated (Saad, 2003). This method is useful for general nonsymmetric matrices. GMRES is the most popular Krylov subspace method applicable to any invertible matrix A.

2.6.2 Bi-Conjugate Stabilized (BI-CGSTAB)

The GMRES method retains orthogonality of the residuals by using long recurrences, at the cost of a larger storage demand. The Bi-Conjugate Gradient method takes another approach, replacing the orthogonal sequence of residuals by two mutually orthogonal sequences (Varga, 1962). The update relations for residuals in the Conjugate Gradient method are augmented in the BiConjugate Gradient method by similar relations.Thus we update two sequences of residuals.Bi-Conjugate requires computing a matrix-vector product and a transpose product . In some applications, the latter product may be impossible to perform, for instance if the matrix is not formed and the regular productis only given as an operation.

2.6.3 QuasiMinimal Residual (QMR)

The BiConjugate Gradient method often displays rather irregular convergence behavior. Moreover, the implicit decomposition of the reduced tridiagonal system may not exist, resulting in breakdown of the algorithm. A related algorithm, the Quasi-Minimal Residual method attempts to overcome this problem. The main idea behind this algorithm is to solve the reduced tridiagonal system in a least squares sense, similar to the approach followed in GMRES. Since the constructed basis for the Krylov subspace is biorthogonal, rather than orthogonal as in GMRES, the obtained solution is viewed as a quasi-minimal residual solution. Additionally, QMR uses look-ahead techniques to avoid breakdowns in the process(Varga, 1962).

2.7 Matrix Splitting

According to Hadjidimos (1987), the first step in the construction of stationary iterative methods usually begins with splitting of matrix A. Thus, A=M – N is a splitting of matrix A.

$$M = D \text{ and } N = -(L + U) \tag{2.12}$$

$$M_{JC} = D \tag{2.13}$$

$$M_{GS} = D + L,\ N = -U. \tag{2.14}$$

$$M_{SOR} = \text{SOR} \quad M = \frac{1}{\omega}(D + L) \tag{2.15}$$

where Dis the diagonal matrix A, L and U are lower and upper triangular matrix M_{JC} , M_{GS} and M_{SOR} are the diagonals for Jacobi, Gauss-Seidel and SOR methods respectively so that Ax=b is equivalent to $x = Tx + C$, where

$$T = M^{-1}N \text{ and } C = M^{-1}b \tag{2.16}$$

2.8Spectral Radius

The spectral radius of an n × n matrix A is$\rho(A) = \max|\lambda| = \lim_{n\to\infty} \|A^n\|^{1/n}$ where λ is an eigenvalue of A. The theory of matrix splitting plays an important role in convergence analysis for the iterative Scheme.The concept of spectral radius allows us to make a complete description of eigenvalues of a matrix andis independent of any particular matrix norm (Kelly, 1995).Consider the linear system Ax = b for the iterative solution of systems of linear equations, it is customary to represent the matrix A as A = M $-$ N: If the matrix M is nonsingular, the iterative method is expressed in the form:

$$x^{(k+1)} = M^{-1}Nx^{k} + M^{-1}b \ ; \quad n \geq 0 : \tag{2.17}$$

Equation (2.17) converges to the unique solution x = A^{-1}bof the system Ax = b for each initial vector $x^{(0)}$ if, and only if, $\rho(M^{-1}N) < 1$, where $\rho(M^{-1}N)$is the spectral radius of the iteration matrix $M^{-1}N$.

2.9Acceleration Schemes

2.9.1Chebyshev Extrapolation Scheme

TheChebyshev Extrapolation is applied to iterative scheme uses the three-term recursion (Hageman and Young 1981) as shown below:

$$\hat{x}^{(k+1)} = Tx^{(k)} + C, \tag{2.18}$$

$$x^{(k+1)} = x^{(k-1)} + \alpha_k\{x^{(k)} - x^{(k-1)} + \beta(\hat{x}^{(k+1)} - x^{(k)})\}, \tag{2.19}$$

where T is iterative matrix, C is the nonsingularmatrix,α_n and β are the extrapolation parametersWrigley (1963) . The formulation of the extrapolation process is preferred since at each iteration of one parameter, alpha need to be calculated. The application of Chebyshev extrapolation in the real eigenvalue case requires the knowledge of two parameters a and b which are respectively the upper and lower bound of the eigenvalue of the iterative matrix.The extrapolation parameters α_n and β are then given in terms of a and b (Arioli *et al*, 1989).

$$\beta = \frac{2}{2-a-b} \tag{2.20}$$

$$\alpha_k = \frac{2\gamma T_n(\gamma)}{T_{n+1}(\gamma)} \text{for } k \geq 1 , \tag{2.21}$$

$$\gamma = \frac{(2-a-b)}{(a-b)} \tag{2.22}$$

$$a = \lambda_{\min}(D^{-1}A) \tag{2.23}$$

$$b = \lambda_{max}(D^{-1}A) \tag{2.24}$$

$$T_n(\gamma) = \cos (n \cos^{-1} \gamma) \tag{2.25}$$

According to Duan and Zheng (2008) when the basic iterationmatrix is similar to a diagonal matrix and all eigenvalues are real and lie in theinterval [a, b], the Chebyshev extrapolation scheme for SOR employs the Chebyshevpolynomials, where D = nonsingular diagonal matrix (A) andλ=eigenvalue of A.An important number of applications in science and engineering require the numerical solution of large nonsymmetric matrix for eigenvalue problem (Saad ,1984) that involved computing the problem of eigenvalue.

Peng and Lin (2012) presented restarted acceleration scheme without using Chebyshev polynomial to work as the acceleration process for stationary iterative methods. Refinement after a fixed number of iteration and the improved approximate was applied using both inner and outer loops to obtain more accurate approximation.

The Chebyshev extrapolation algorithm is as follows:

1. Input the coefficient matrix A, initial guess x^0 and the RHS vector b and set maximum value for iteration

2. Set the tolerance, $\varepsilon = 0.0001$
3. Compute the residual vector $\mathrm{r} = \mathrm{b} - \mathrm{Ax}^0$
4. Compute $\mathrm{z} = \frac{\mathrm{M}}{\mathrm{r}}$
5. Compute the error vector
6. Stop if error $<\varepsilon$

2.9.2 Accelerated Gradient Scheme

Accelerate Gradient acceleration scheme improves the convergence stationary iterative methods for solving a large and sparse system.Accelerated gradient schemes applied the idea of initial vector for the iterative scheme (Auslender and Teboulle, 2006).

$$x^{(k+1)} = y^k - \nabla f(y^k) \tag{2.26}$$

$$y^{(k+1)} = x^{(k+1)} + \beta\left(x^{(k+1)} - x^{(k)}\right) \tag{2.27}$$

$$\beta = \frac{1 - \sqrt{\frac{a}{b}}}{1 + \sqrt{\frac{a}{b}}} , \tag{2.28}$$

where∇f is the gradient function,a and b are the largest and smallest eigenvalues of matrix A.

The Accelerated gradient algorithm is as follows:

1. Let $x^0 \epsilon R^n$ and $y^0 = x^0$ and θ^0=1 k$\geq$ 0
2. Compute x^{k+1}= $y^k - \nabla f(y^k)$
3. Compute θ^{k+1}
4. Compute $y^{k+1} = x^{k+1} + \theta^{k+1}(x^{k+1} - x^k)$

2.9.3Residual Smoothing Scheme

An iterative method for solving a linear system Ax = b produces iterates x_k with associated residualnorms that, in general, need not decrease "smoothly" to zero. "Residual smoothing" techniques are considered that generate a second sequence (Walker and Zhou, 1994). Let $\{x_k\}$ be a sequence of approximate solutions of a linearsystem A x = b, A $\in \mathrm{R}^{(n\times n)}$, and let $\{r_k \equiv$ b − A$x_k\}$ be the associated sequence of residuals. We consider the following general residual smoothing technique:Let $y_0 = x_0$, s_0 = r_0,k=1,2,3…….

$$x^k = \mathrm{M} + (\mathrm{N}x + \mathrm{b}) \qquad (2.29)$$

$$y_k = y_{k-1} + (x_k - y_{k-1}) \qquad (2.30)$$

$$s_k = s_{k-1} + x^k(r_k - s_{k-1}) \qquad (2.31)$$

The Residual Smoothing Algorithm is as follows:

1. Let $s^0 = r^0$ and $y^0 = x^0$
2. Compute $x^k$and r^k k= 1,2………..
3. Compute $d^k = \frac{s^T_{k-1}(r^k - s^{k-1})}{\|r^k - s^{k-1}\|}$
4. Find x^k= M + (Nx^0 + b)
5. Set $s^k = s^{k-1} + d^k(r^k - s^{k-1})$
6. Compute $y^k = y^{k-1} + d^k(x^k - y^{k-1})$

2.9.4 Richardson Extrapolation Scheme

The Richardson Extrapolation Scheme is of the form:

$$x^{(k+1)} = (\mathrm{I} - \mathrm{T})x^k + \mathrm{C} \qquad (2.32)$$

since A= M− N, T=M^{-1}N and C = M^{-1}b, where M and N are nonsingular matrices.

(i) The iterative scheme (Eqn 2.31) leads to the Richardson extrapolation scheme:

$$x^{(k+1)} = \big((1-\omega)I + \omega T\big)x^k + \omega C \qquad (2.33)$$

where ω and I are the relaxation parameters and identity matrix respectively.The Richardson Algorithm is as follows:

1. Input the coefficient matrix A, initial guess x^0 and the RHS vector b.
2. Form an approximate solution $x^0, x^1, x^2 \dots\dots\dots x^k \dots\dots\dots$ $k \geq 0$
3. Find the output vector $r = b - Ax^k$ and add this vector to x^k to obtain x^{k+1} $= x^k + (b - Ax^k)$
4. Add input error vector $x - x^0$ to x^k
5. Compute $x^{(k+1)} = \big((1-\omega)I + \omega T\big)x^k + \omega C$

2.10 Relative Residual

The accuracy, stability and fast convergence of a solution to systems of linear equations are determined by residual as well as relative residual. Consider solution of the linear system Ax = b, with exact answer x and computed solution x^*. Thus, we expect an error e = $x - x^*$. Since x is not known to us in general we often judge the accuracy of the solution with assumption that a small residual may guarantee a small error.So large relative residual implies large backward error in the matrix and algorithm used to compute solution is unstable. Another way of saying this is that a stable algorithm will invariably produce a solution with small relative residual, irrespective of the conditioning of the problem, and hence a small residual by itself, sheds little light on the quality of the approximate solution. Therefore the residual is related to relative residual as follows:

Relative Residual (Relres) $= \dfrac{\|A\,x - Ax^*\|}{\|A\,x\|} = \dfrac{\|b - b^*\|}{\|b\|} = \dfrac{\|\,r\,\|}{\|\,b\,\|}$ (2.34)

There are several stopping criteria which one may apply to iterative systems. Our stopping criterion was to use the relative residual: $\frac{\|A\,x - Ax^{*}\|}{\|A\,x\|} < \varepsilon$ where is taken as 10^{-4}.

2. 11 Related Works

Kalambi (2008) studied the three stationary iterative methods (Jacobi,Gauss-Seidel and SOR)for the solution of linear equations and his results shows that the SOR method was more efficient than the other two iterative methods, considering their performance, using parameters as time to converge, number of iterations required to converge, storage and level of accuracy.

Zhang *et al.*, (2014) studied the three stationary iterative methods together withAlternating Direction Implicit, Chebyshev and Conjugate gradient method.They considered Chebyshev and Conjugate gradient methods as effective.

Santo and Linhares, (1986) proved the sufficient condition for the convergence of Chebyshev acceleration methods applied to numerical solution of algebraic linear system. The convergence condition depends on the eigenvalues of a particular matrix.

Unified backward iterative matrix was proposed by (Wang *et al.*, 2014)to solve large scale linear systems and backward and Jacobi iteration algorithms are employed.Hadjidimos (1978) proposed two-parameter generalization of the SOR method known as Accelerated Overrelaxation (AOR) method using the stationary iterative scheme and matrix splitting method.Peng and Lea (2012) studied an acceleration scheme based on stationary iterative methods for solving linear system of equations. They used a wide range of Chebyshev-like polynomials for the accelerating process without estimating the bounds of the iterative matrix. Chebyshev extrapolation can be used to accelerate the convergence of Jacobi method (Young and Jea, 1980).

Iqbal, (2012) investigated the use of projection techniques and proposed the residual iterative method which minimized the norm of the residual over Krylov subspace.Further, Young and Jea, (1980) also proposed three generalization of Conjugate Gradient acceleration methods which are designed to speed up the convergence of stationary iterative methods.

Walker and Zhou (1994) studied residual smoothing techniques for solving systems of linear equations in residual norm sequence produced by certain pairs of Krylov subspace methods and applied minimal residual smoothing to an orthogonal residual methodresults in a minimal residual method.

From the above discussion, researchers have dealt with numerous areas in iterative methods for the solution of SLE. However,the convergence of Jacobi Gauss-Seidel and SOR methods need to be improved using acceleration schemes. That is to say, incorporation of the named iterative methods into the identified acceleration schemes using the selected linear systems. Therefore, this study focused on acceleration schemes to assess the solution of SLE.

2. 12 Conclusion

This chapter detailed the sources of systems of linear equations, concepts of stationary iterative

Methods and their convergence: It also explained the concepts of the identified acceleration schemes and their algorithm and finally related works of stationary and accelerationschemes.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This section presents details of the experimental work which includes a discussion of the stationary iterative methods incorporated into the acceleration schemes and other procedures employed to accomplish the study objectives. .

3.1Stationary Iterative Method with Chebyshev Extrapolation Scheme

The Jacobi, Gauss-Seidel and SOR are incorporated into Chebyshev extrapolation schemes. The incorporation depends on the nonsingular matrix M found in theiterative matrix T = $M^{-1}N$(see eqn2.16). The Matrix M, wasputin the Chebyshev extrapolation algorithm (see Eqn2.12) as follow:

$$Z = \frac{M}{r}, \tag{3.1}$$

Wherer is the residualof $Ax = b$. Eqn (3.1) with respect to Jacobi, Gauss-Seidel and SOR methods become:

$$Z = \frac{M_{JC}}{r}\text{(for Jacobi)}, \tag{3.2}$$

$$Z = \frac{M_{GS}}{r}\text{(for Gauss-Seidel)}, \tag{3.3}$$

$$Z = \frac{M_{SOR}}{r}\text{(for SOR)}, \tag{3.4}$$

where M_{JC}, M_{GS} and M_{SOR} are as defined in Eqns (2.13, 2.14 and 2.15) respectively.

The next step is to run the coded version of the algorithm of the Chebyshev extrapolation scheme using each of the eqns (3.2 to 3.4) as required.

3.2Stationary Iterative Method with Richardson Extrapolation Scheme.

The nonsingular matrix M in the iterative matrix T = $M^{-1}N$ was put in the Richardson Extrapolation Scheme (eqn 2.33) and obtained:

$$T = {M_{JC}}^{-1}N \text{(for Jacobi)}, \tag{3.5}$$

$$T = {M_{GS}}^{-1}N \text{(for Gauss-Seidel)}, \tag{3.6}$$

$$T = {M_{SOR}}^{-1}N \text{(for SOR)}, \tag{3.7}$$

where M_{JC}, M_{GS} and M_{SOR} are as defined in Eqns (2.13, 2.14 and 2.15) respectively.

The next step is to run the coded version of the algorithm of the Richardson extrapolation scheme using each of the eqns (3.5 to 3.7) as required

3.3Stationary Iterative Method with Residual Smoothing Scheme

The nonsingular matrix M in eqn (2.29) was replaced with M_{JC} (eqn 2.11),M_{GS}(eqn 2.12) and M_{SOR}(eqn 2.13) and obtained:

$$x_n = M_{GS} + (Nx + b), \tag{3.8}$$

$$x_n = M_{GS} + (Nx + b), \tag{3.9}$$

$$x_n = M_{SOR} + (Nx + b), \tag{3.10}$$

where M_{JC}, M_{GS} and M_{SOR} are as defined in Eqns (2.13, 2.14 and 2.15)) respectively.

The next step is to run the coded version of the algorithm of the Residual scheme using each of the eqns (3.8 to 3.10) as required.

3.4Stationary Iterative Method with Accelerated Gradient Scheme

For this scheme, eqn (2.26) was replaced with:

$$x^{(k+1)}= D^{-1}[b-(L+U)x^{k}] \qquad (3.11)$$

Now, substituting M= D from (eqn 2.12) and eqn (3.11) and obtained

$$x^{(k+1)}= \mathrm{M}^{-1}[b-(L+U)x^{k}] \qquad (3.12)$$

Again, replacing M in eqn (3.12) and obtained:

$$x^{(k+1)}= \mathrm{M_{JC}}^{-1}[b-(L+U)x^{k}] \qquad (3.13)$$

$$x^{(k+1)}= \mathrm{M_{GS}}^{-1}[b-(L+U)x^{k}] \qquad (3.14)$$

$$x^{(k+1)}= \mathrm{M_{SOR}}(^{-1}[b-(L+U)x^{k}] \qquad (3.15)$$

where M_{JC}, M_{GS} and M_{SOR} are as defined in Eqns (2.13, 2.14 and 2.15) respectively.

The next step is to run the coded version of the algorithm of the Accelerated gradient scheme using each of the eqns (3.13 to 3.15) as required.

3.5 Experimental work

This section discussed the construction, processing and solutions of the Tridiagonal, banded and Dense SPD linear systems.

3.5.1 Problem Construction

The identified SLE were constructed using the systems obtained from the available source and Matlab code generation). The coefficient matrix A of the problem Ax = b for each Tridiagonal, Banded and Dense SPD were randomly generated using Matlab code (see Appendix D, E and F).The solution vector x, for each generated system is x=

$(1,1 \ldots \ldots \ldots . n)^{\mathrm{T}}$ where n is the number of variables. The RHS vector b, was then obtained as $\mathrm{b} = Ax$ for each n by n square matrix. The problem construction for each Tridiagonal, Banded and Dense SPD was started using n = 9, 25, 50,100, 200, 500 to 800. The size of the systems was varied to assess the performance of each method.

3.5.2Problem Processing

In order to process the identified linear systems, the Matlab codes for Jacobi, Gauss-Seidel and SORand the acceleration schemes for each specific system were entered into the M-File editor of the Matlab software. The coefficient matrix A and RHS vector for a particular system were put into the code to be computed. The results were then displayed at the command window where each of the named performance criteria was also recorded.

3.5.3 Problem Solution

In the solution of the named SLE, the performance criteria considered were convergence, number of iterations, speed of convergence (in seconds) and relative residual.Tridiagonal systems (Appendix F) was first computed using Jacobi, Gauss-Seidel and SOR iterative methods with n= 9 to assess the identified performance criteria. The next computation dealt with finding the solution of the incorporated Jacobi, Gauss-Seidel and SOR into the identified accelerations schemes. That is solving the system using Chebyshev extrapolation scheme(CES) with Jacobi, Chebyshev extrapolation scheme (CES) with Gauss-Seidel,Chebyshev extrapolation scheme (CES) with SOR, Accelerated gradient scheme (RES) with Jacobi, Accelerated gradient scheme (RES) with Gauss-Seidel,Accelerated gradient scheme (RES) with SOR, Residual smoothing(RS) with Jacobi,Residual smoothing(RS) with Gauss-Seidel,Residual smoothing(RS) with SOR, Richardson extrapolation scheme (RES) with Jacobi, Richardson extrapolation scheme (RES) with Gauss-Seidel and Richardson extrapolation scheme (RES). Then,the following known iterative methods (Richardson, Chebyshev, GMRES, MINRES,QMR and BICGSTAB) were also applied to solve the Tridiagonal system to assess and compare the above named performance criteria.

Furthermore, the size of the Tridiagonal system was then increased to n=25 which was solved using Jacobi, Gauss-Seidel, SOR methods first followed by the acceleration schemes with the stationary iterative methods (CES with Jacobi, CES with Gauss-Seidel, CES with SOR, RES with Jacobi, RES with Gauss-Seidel,RES with SOR, RS with Jacobi, RS with Gauss-Seidel, RS with SOR, AGS with Jacobi, AGS with Gauss-Seidel,AGS with SOR) as well as the known iterative methods. The above procedures were repeated with varying size of Tridiagonal system (n=50,100,200,500 and 800).

Moreover, having dealt with Tridiagonal system, the next chosen system was Dense and SPD. Again, some real life problems are dense which are at the same time having special property (SPD).The computation of dense SPD started with n=9 as used in the previous procedures using the stationary iterative methods with the acceleration schemes and then applying the known iterative methods to assess the same performance criteria.

Again, the next system considered was Banded started with n=9 which was solved using Jacobi, Gauss-Seidel, SOR methods followed by the acceleration schemes with the stationary iterative methods (CES with Jacobi, CES with Gauss-Seidel, CES with SOR, RES with Jacobi, RES with Gauss-Seidel,RES with SOR, RS with Jacobi, RS with Gauss-Seidel, RS with SOR, AGS with Jacobi, AGS with Gauss-Seidel,AGS with SOR) and finally the identified acceleration methods.

Finally, relaxation parameterω ranging between $0 < \omega < 2$ was used for SOR, Richardson extrapolation scheme and Acceleration gradient schemes respectively. When solving system, the parameter was arbitrarily selected and varied for each iteration within the given range so optimal relaxation parameter for each particular system could be obtained. The significance of this optimal relaxation parameter is that it helps to reduce the number of iterations. The smaller the number of iterations, the lesser the amount of data being stored in a computer and vice versa and the faster the speed.

3.6 Conclusion

This chapter explained the incorporation of the stationary iterative methods into the identified acceleration schemes. It also showed the processes of the experimental works for the identified systems of linear equations.

.

CHAPTER FOUR

NUMERICAL EXPERIMENTS AND RESULTS

4.0 Introduction

In this section, we present some numerical results to show the performance acceleration schemes on the identified linear systems.We ran our algorithm using Matlab software version 7.0.1.

4.1Numerical Results

The tables below display the results of the identified linear system with varying sizes and the performance criteria. The relaxation parameters for each method were placed against the specified methods. Flag value of "0" indicates that the method applied converged to a solution and "1" showed that the method did not convergence for a specific number of iterations.Again, Flag value of "2" indicate preconditioned was ill-conditioned, "3" means method stagnated(two consecutive iterations were the same) and "4" indicate one of the scalars is too large to continue the computation.

4.2 Tridiagonal System

This section explained the results of Tridiagonal systems of linear equation. The experimental works begun with different dimensions of the systems. The size of Tridiagonal systems started with n = 9, and then increased to 25, 50, 100, 200, 500 and finally 800. The tables on the next page elaborated the the numerical results.

Table 4.1:Tridiagonal System n=9

ITERATIVE METHODS n= 9				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	28	0.010000	2.1129e-005
Gauss-Seidel	0	16	0.010000	0.250000
SOR ω=1.12	0	11	0.025000	0.250000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	17	0.045000	6.0578e-004
CES with GS	0	12	0.030000	8.7898e-004
CES with SOR ω=0.60	0	10	0.050000	5.0506e-004
RES with Jacobi ω=1.12	0	29	0.042000	1.3590e-005
RES with GS ω=1.0	1	101	0.030000	5.2242e+008
RES with SOR ω=1.20	1	101	0.040000	5.4020e+016
AGS with Jacobi	0	6	0.005000	4.6060e-016
AGS with Gauss-Seidel	0	6	0.010000	4.6060e-016
AGS with SOR ω= 0.9	0	6	0.005000	4.6060e-016
RS with Jacobi	0	2	0.006000	2.2480e-016
RS with Gauss-Seidel	0	2	0.005000	4.8115e-016
RS with SOR ω=1.13	0	2	0.005000	5.8376e-016
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.000082	1.00000
Richardson Acceleration	0	3	0.005000	1.1305e-016
Gmres	0	6	0.132780	7.5e-016
Minres	0	9	0.002641	1.8e-005
Qmr	0	6	0.002641	9.3e-016
Bicgstab	0	5.5	0.093143	1.6e-016

From Table 4.1, all the methods converged to an approximate solutionwith the exception of RESwith Gauss-Seidel and RES with SOR and this can be observedfrom their relative residuals. The relative residual conformed with the convergence of the methods except RES with Gauss-Seidel and RES with SOR because of the its large value (5.2242e+008 and 5.4020e+016).This indicates that the corresponding schemes failed to converge. TheRS and Chebyshev acceleration methods have the fastest convergence in terms of number of iterations.The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.6 and 1.2 respectively.

Table 4.2: Tridiagonal System n= 25

ITERATIVE METHODS n= 25				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	11	0.052000	9.0872e-005
Gauss-Seidel	0	7	0.031000	0.2500
SOR ω = 1.11	0	6	0.078000	0.2000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	3	0.015000	6.8408e-004
CES with GS	0	7	0.031000	4.6425e-004
CES with SOR ω =1.01	0	7	0.063000	9.8979e-004
RES with Jacobi ω =1.12	0	14	0.043000	5.1065e-004
RES with GS ω =1.25	0	8	0.024000	5.5660e-004
RES with SOR ω =1.11	0	11	0.016000	5.8001e-004
AGS with Jacobi	0	9	0.020000	5.7987e-007
AGS with Gauss-Seidel	0	9	0.045000	5.2208e-007
AGS with SOR ω = 0.9	0	9	0.030000	5.1255e-007
RS with Jacobi	0	2	0.025000	3.4609e-007
RS with Gauss-Seidel	0	2	0.030000	2.8579e-007
RS with SOR ω = 1.13	0	2	0.027000	2.0754e-007
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.063000	2.7519e-016
Richardson Acceleration	0	3	0.125000	9.9403e-008
Gmres	0	9	0.031000	5.1e-007
Minres	0	8	0.109000	5.5e-007
Qmr	0	9	0.031000	4.4e-007
Bicgstab	0	5.5	0.109000	2.4e-007

From Table 4.2, all the methods (stationary iterative methods and the accelerated schemesconverged. For this type of problem RS and Chebyshev acceleration methods converged faster than all the other methods in terms of number of iteration. The minimum and maximum optimal relaxation parameters for the stationary iterative methods and acceleration schemes are 0.9 and 1.25 respectively. For this problem only the acceleration methods, RS scheme and CES with Jacobi improved the convergence of the stationary iterative methods.The relative residuals for each methods are relatively small.

Table 4.3: Tridiagonal System n=50

ITERATIVE METHODS n= 50				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	11	0.015000	9.8114e-005
Gauss-Seidel	0	8	0.095000	0.2000
SOR $\omega = 1.01$	0	8	0.062000	0.2000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	3	0.062000	6.3410e-004
CES with GS	0	4	0.156000	8.1900e-004
CES with SOR $\omega =0.9$	0	4	0.125000	7.6853e-004
RES with Jacobi $\omega =$	0	14	0.013900	3.8612e-005
RES with GS $\omega =1.11$	0	12	0.023000	1.2545e-005
RES with SOR $\omega =0.9$	0	11	0.023200	1.6726e-005
AGS with Jacobi	0	9	0.230000	5.1450e-007
AGS with Gauss-Seidel	0	9	0.253000	2.1452e-007
AGS with SOR $\omega = 0.9$	0	9	0.212300	3.1312e-007
RS with Jacobi	0	2	0.075000	1.00000
RS with Gauss-Seidel	0	2	0.012000	1.00000
RS with SOR $\omega = 1.01$	0	2	0.311200	1.00000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.015000	2.1379e-015
Richardson Acceleration	0	3	0.017000	1.6651e-007
Gmres	0	9	0.015000	5.9e-007
Minres	0	8	0.015000	5.9e-007
Qmr	0	9	0.016000	5.9e-007
Bicgstab	0	5.5	0.015000	2.6e-007

From Table 4.3, all the methods (stationary iterative methods, the accelerated schemes and acceleration methods converged. For this type of problem RS and Chebyshev acceleration methods converged faster than all the other methods in terms of number of iteration and RES did not improve the stationary iterative methods.The minimum and maximum optimal relaxation parameters for the stationary iterative methods and acceleration schemes are 0.9 and 1.11respectively.

Table 4.4 : Tridiagonal System n=100

ITERATIVE METHODS n= 100				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	12	0.015000	4.0542e-005
Gauss-Seidel	0	8	0.077000	0.2000
SOR ω = 1.01	0	8	0.093000	0.2000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	3	0.156000	5.7303e-004
CES with GS	0	4	0.343000	8.3607e-004
CES with SOR ω =1.22	0	4	0.562000	7.8390e-004
CES with Jacobi ω =0.9	0	14	0.132000	3.9974e-005
RES with GS ω =1.01	0	13	0.120000	2.6125e-005
RES with SOR ω =1.25	0	9	0.320000	1.6455e-005
RES with Jacobi	0	9	0.328000	5.7652e-007
AGS with Gauss-Seidel	0	9	0.114000	3.1443e-007
AGS with SOR ω = 1.01	0	9	0.013000	3.1451e-007
RS with Jacobi	0	2	0.119000	1.00000
RS with Gauss-Seidel	0	2	0.023000	1.00000
RS with SOR ω = 0.9	0	2	0.130000	1.00000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.047000	2.0106e-015
Richardson Acceleration	0	3	0.023000	1.4708e-007
Gmres	0	9	0.016000	5.9e-007
Minres	0	8	0.015000	5.9e-007
Qmr	0	9	0.015000	5.9e-007
Bicgstab	0	5	0.016000	8.4e-007

From Table 4.4, all the methods converged and CES with Jacobi and RES with Gauss-Seidel did not improve the stationary iterative methods. The minimum and maximumoptimal relaxation parameters for the stationary iterative methods and acceleration schemes are 0.9 and 1.22 respectively.

Table 4.5: Tridiagonal System n=200

ITERATIVE METHODS n= 200				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	12	0.047000	4.1249e-005
Gauss-Seidel	0	9	0.135000	0.2000
SOR ω = 1.01	0	8	0.328000	0.2000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	3	0.858000	4.9097e-004
CES with GS	0	6	1.357000	8.4722e-004
CES with SOR ω =1.01	0	4	1.513000	7.9387e-004
RES with Jacobi ω =1.11	0	15	1.320000	1.8680e-005
RES with GS ω =0.9	0	13	2.100000	3.6345e-005
RES with SOR ω =0.9	0	10	1.020000	1.6723e-005
AGS with Jacobi	0	9	0.280000	5.9057e-007
AGS with Gauss-Seidel	0	9	0.253000	4.3212e-007
AGS with SOR ω = 1.01	0	9	0.013000	3.2152e-007
RS with Jacobi	0	2	0.382000	1.00000
RS with Gauss-Seidel	0	2	0.350000	1.00000
RS with SOR ω = 1.11	0	2	0.210000	1.00000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.312000	2.1717e-015
Richardson Acceleration	0	3	0.022000	1.8803e-007
Gmres	0	9	0.015000	6.1e-007
Minres	0	8	0.015000	6.1e-007
Qmr	0	9	0.015000	6.1e-007
Bicgstab	0	5.5	0.015000	2.6e-007

From Table 4.5, RES with Jacobi,RES with Gauss-Seidel and did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.11 respectively.

Table 4.6: Tridiagonal System n=500

ITERATIVE METHODS n= 500				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	11	0.062000	1.0420e-004
Gauss-Seidel	0	9	0.370000	0.2000
SOR ω = 1.01	0	9	1.903000	0.2000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	3	4.868000	5.0792e-004
CES with GS	0	3	6.453000	7.3454e-004
CES with SOR ω =0.9	0	2	9.376000	0.0097
RES with Jacobi ω =1.01	0	15	3.200000	4.1031e-005
RES with GS ω =1.11	0	12	5.202000	4.3033e-005
RES with SOR ω =0.9	0	6	3.230000	3023e-005
AGS with Jacobi	0	9	2.137000	5.5644e-007
AGS with Gauss-Seidel	0	9	2.500000	2.2452e-007
AGS with SOR ω =	0	9	3.021000	2.1851e-007
RS with Jacobi	0	2	2.823000	1.00000
RS with Gauss-Seidel	0	2	3.100000	1.00000
RS with SOR ω = 0.9	0	2	1.950000	1.00000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	1.685000	2.2830e-015
Richardson Acceleration	0	3	0.071000	1.7919e-007
Gmres	0	9	0.016000	5.8e-007
Minres	0	8	0.016000	5.8e-007
Qmr	0	9	0.031000	5.8e-007
Bicgstab	0	5.5	0.015000	2.4e-007

From Table 4.6, RES with Jacobi,RES with Gauss-Seideldid not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.11 respectively.

Table 4.7: Tridiagonal System n=800

ITERATIVE METHODS n= 800				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	13	0.172000	1.6705e-005
Gauss-Seidel	0	9	0.821000	0.2000
SOR ω = 0.9	0	6	6.443000	0.2000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	3	19.297000	2.6345e-005
CES ev with GS	0	4	30.982000	8.7189e-004
CES with SOR ω =0.9	0	4	34.242000	1.6265e-005
RES with Jacobi ω =1.11	0	6	20.34400	1.6305e-005
RES with GS ω =1.12	0	5	23.54300	1.5305e-005
RES with SOR ω =0.9	0	4	13.459300	2.6715e-005
AGS with Jacobi	0	9	5.148000	5.8790e-007
AGS with Gauss-Seidel	0	9	5.350000	3.1352e-007
AGS with SOR ω = 0.9	0	9	5.021000	4.1342e-007
RS with Jacobi	0	2	7.676000	1.00000
RS with Gauss-Seidel	0	2	6.453000	1.00000
RS with SOR ω = 0.9	0	2	5.343000	1.00000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	8.034000	3.1274e-015
Richardson Acceleration	0	3	0.219000	1.6659e-007
Gmres	0	9	0.047000	5.6e-007
Minres	0	8	0.047000	5.7e-007
Qmr	0	9	0.078000	5.7e-007
Bicgstab	0	5.5	0.062000	2.2e-007

From Table 4.7,The speed of the CES and RES were quite slow as compare to the remaining methods but the relative residual are sufficiently small. The minimum and maximum optimal relaxation parameters for the stationary iterative methods and acceleration schemes are 0.9 and 1.12 respectively.

4.3 Dense SPD Systems.

This section gave the results of Dene SPD systems of linear equations.

Table 4.8: Dense SPD System n = 9

ITERATIVE METHODS n= 9				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	1	100	0.156000	1.5563e+023
Gauss-Seidel	0	6	0.047000	0.0825
SOR ω= 0.9	0	5	0.062000	0.0825
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	8	0.094000	3.0967e-004
CES with GS	0	7	0.031000	9.3008e-004
CES with SOR ω=1.01	0	7	0.016000	0.0034
RES with Jacobi ω=1.25	1	101	0.067000	4.9336e+037
RES with GS ω=1.01	1	101	0.047000	4.5069e+013
RES with SOR ω=1.11	1	101	0.072100	8.0130e+012
AGS with Jacobi	0	5	0.031000	3.1592e-007
AGS with Gauss-Seidel	0	5	0.015000	3.1592e-007
AGS with SOR ω= 0.9	0	5	0.016000	3.1592e-007
RS with Jacobi	0	2	0.031000	4.8449e-007
RS with Gauss-Seidel	0	2	0.016000	4.3684e-007
RS with SOR ω=1.25	0	2	0.015000	3.8658e-007
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	2.7147e-016
Richardson Acceleration	0	3	0.031000	3.0564e-008
Gmres	0	5	0.015000	3.2e-007
Minres	0	4	0.124000	3.2e-007
Qmr	0	5	0.124000	3.2e-007
Bicgstab	0	3.5	0.358000	2.4e-008

From Table 4.8, Jacobi, RES with Jacobi,RES withGauss-Seideland RES with SOR did not converge which corresponds with the largest values of the relative residual.The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.9 : Dense SPD System n = 25

ITERATIVE METHODS n= 25				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	95	0.952000	1.0182e-004
Gauss-Seidel	0	7	0.091000	0.0250
SOR ω= 0.9	0	6	0.063000	0.0250
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	5	0.093000	1.2176e-004
CES with GS	0	5	0.063000	1.0908e-004
CES with SOR ω=0.9	0	4	0.031000	9.9137e-004
RES with Jacobi w=0.6	0	8	0.054000	6.0392e-005
RES with GS ω=0.6	0	8	0.013000	7.7639e-005
RES with SOR ω=0.6	0	5	0.029000	9.5900e-005
AGS with Jacobi	0	4	0.088000	5.6974e-007
AGS with Gauss-Seidel	0	4	0.045000	5.6974e-007
AGS with SOR w= 0.6	0	4	0.018000	5.6974e-007
RS with Jacobi	0	2	0.046000	1.000000
RS with Gauss-Seidel	0	2	0.032000	1.000000
RS with SOR ω=1.23	0	2	0.031000	8.5985e-007
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	6.5446e-017
Richardson Acceleration	0	2	0.047000	1.0895e-007
Gmres	0	4	0.016000	5.7e-007
Minres	0	3	0.078000	5.7e-007
Qmr	0	4	0.016000	5.7e-007
Bicgstab	0	2.5	0.094000	4.6e-008

From Table 4.9, all the methods improved the stationary iterative methods except RES with Jacobi, RES with Jacobi, RES with Gauss-Seidel and RES with SOR. This is confirmed by the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.10 : Dense SPD System n = 50

ITERATIVE METHODS n= 50				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	1	1000	10.374000	8.5170e+264
Gauss-Seidel	0	11	0.078000	0.0250
SOR ω = 0.9	0	9	0.031000	0.0250
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	7	0.055000	2.9257e-004
CES with GS	0	7	0.093000	2.5571e-004
CES with SOR ω =0.9	0	6	0.156000	9.2620e-004
RES with Jacobi ω =1.12	1	101	2.120000	7.4506e+033
RES with GS ω =1.01	1	89	1.320000	1.6705e+135
RES with SOR ω =0.9	0	34	0.230000	1.6311e-005
AGS with Jacobi	0	4	0.031000	7.8200e-007
AGS with Gauss-Seidel	0	4	0.023000	2.3433e-007
AGS with SOR ω = 0.9	0	3	0.012000	2.1033e-007
RS with Jacobi	0	2	0.015000	1.2275e-004
RS with Gauss-Seidel	0	2	0.016000	1.1542e-004
RS with SOR ω = 1.25	0	2	0.031000	1.3364e-004
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	7.1411e-017
Richardson Acceleration	0	3	0.004000	2.5505
Gmres	0	4	0.004000	7.8e-007
Minres	0	3	0.002000	7.8e-007
Qmr	0	4	0.008000	7.8e-007
Bicgstab	0	2.5	0.003000	1e-007

From Table 4.10, Jacobi, RES with Jacobi,RES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.11: Dense SPD System n= 100

ITERATIVE METHODS n= 100				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	1	500	6.411000	3.6534e+285
Gauss-Seidel	0	20	0.327000	0.0250
SOR ω = 0.9	0	18	0.187000	0.0250
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	11	0.193000	3.9296e-004
CES with GS	0	11	0.375000	4.2941e-004
CES with SOR ω =0.9	0	10	0.390000	2.6847e-004
RES with Jacobi ω =1.01	1	100	0.015000	1.6659e+051
RES with GS ω =0.9	0	23	0.002341	1.2325e-005
RES with SOR ω =1.25	0	12	0.012000	1.2045e-005
AGS with Jacobi	0	5	0.094000	7.6379e-008
AGS with Gauss-Seidel	0	4	0.034000	6.5479e-008
AGS with SOR ω = 1.01	0	4	0.024000	5.6374e-008
RS with Jacobi	0	2	0.015000	1.2575e-004
RS with Gauss-Seidel	0	2	0.016000	1.2642e-004
RS with SOR ω = 1.25	0	2	0.021000	1.5343e-004
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.062000	2.7343e-016
Richardson Acceleration	0	3	0.031000	1.6038e-008
Gmres	0	5	0.005000	7.6e-008
Minres	0	4	0.012000	7.6e-008
Qmr	0	5	0.012000	7.6e-008
Bicgstab	0	2.5	0.007000	1.9e-007

From Table 4.11, Jacobi, RES with Jacobi, did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.12: Dense SPD System n = 200

ITERATIVE METHODS n= 200				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	1	300	5.850000	1.3044e+262
Gauss-Seidel	0	48	0.717000	0.0250
SOR ω = 0.9	0	40	0.718000	0.0250
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	20	0.846000	9.1197e-004
CES with GS	0	17	1.373000	7.4509e-004
CES with SOR ω =0.9	0	19	1.623000	9.7546e-004
RES with Jacobi ω =1.01	1	501	1.782000	6.1542e+206
RES with GS ω =1.25	1	458	1.240000	1.6705e+320
RES with SOR ω =1.20	0	25	0.201000	1.6705e-005
AGS with Jacobi	0	5	0.437000	1.2437e-007
AGS with Gauss-Seidel	0	5	0.013000	1.5347e-007
AGS with SOR ω = 1.01	0	4	0.432000	1.2687e-007
RS with Jacobi	0	2	0.015000	1.4375e-006
RS with Gauss-Seidel	0	2	0.016000	1.3222e-006
RS with SOR ω = 0.9	0	2	0.024000	1.3626e-006
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.250000	2.9463e-016
Richardson Acceleration	0	3	0.047000	3.8526e-008
Gmres	0	5	0.015000	1.5e-007
Minres	0	4	0.016000	1.5e-007
Qmr	0	5	0.015000	1.5e-007
Bicgstab	0	2.5	0.016000	3.3e-007

From Table 4.12, Jacobi, RES with Jacobi,RES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.13: Dense SPD System n =500

ITERATIVE METHODS n= 500				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	1	600	5.179000	3.1852e+254
Gauss-Seidel	0	205	5.725000	0.0250
SOR ω = 0.9	0	167	6.458000	0.0250
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	98	8.425000	0.0069
CES with GS	0	157	15.928000	0.0010
CES with SOR ω =1.01	0	156	17.082000	9.9775e-004
RES with Jacobi ω =0.9	1	502	12.450000	4.2852e+132
RES with GS ω =1.01	1	434	13.030000	1.3405e+235
RES with SOR ω =0.9	0	56	1.0303000	1.6205e+245
AGS with Jacobi	0	5	2.574000	2.74854e-007
AGS with Gauss-Seidel	0	5	4.023000	1.24587e-007
AGS with SOR ω = 1.01	0	4	3.230000	1.54687e-007
RS with Jacobi	0	2	0.013000	1.26345e-006
RS with Gauss-Seidel	0	2	0.016000	1.51232e-006
RS with SOR ω = 1.01	0	2	0.013000	1.52254e-006
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	1.763000	4.1044e-016
Richardson Acceleration	0	3	0.078000	8.2364e-008
Gmres	0	5	0.031000	3.7e-007
Minres	0	4	0.015000	3.7e-007
Qmr	0	5	0.063000	3.7e-007
Bicgstab	0	2.5	0.046000	7.4e-007

From Table 4.13, Jacobi, RES with Jacobi,RES with Gauss-Seideland RES with SOR did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.01 respectively.

Table 4.14: Dense SPD System n = 800

ITERATIVE METHODS n= 800				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	1	200	6.567000	2.2079e+295
Gauss-Seidel	0	466	30.451000	0.0250
SOR ω = 1.11	0	124	13.744000	0.0100
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	556	84.755000	0.02400
CES with GS	0	392	98.780000	9.9281e-004
CES with SOR ω =1.01	0	291	101.182000	9.9904e-004
RES with Jacobi ω =1.25	1	678	76.65000	3.1852e+254
RES with GS ω =1.11	1	457	68.03400	1.5087e+231
RES with SOR ω =0.9	0	26	1.023000	1.5087e-004
AGS with Jacobi	0	5	6.006000	3.7428e-007
AGS with Gauss-Seidel	0	5	7.230000	1.5527e-007
AGS with SOR ω = 1.01	0	4	2.310000	1.3247e-007
RS with Jacobi	0	2	10.265000	1.0000
RS with Gauss-Seidel	0	2	23.759000	1.0000
RS with SOR ω = 1.13	0	2	27.846000	1.0000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	5.585000	5.5581e-016
Richardson Acceleration	0	3	0.156000	37.7154
Gmres	0	5	0.016000	2.7e-007
Minres	0	4	0.011000	2.7e-007
Qmr	0	5	0.017000	2.7e-007
Bicgstab	0	2.5	0.013000	6e-007

From Table 4.14, Jacobi, RES with Jacobi,RES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

4.4 Banded Systems.

This section shows the numerical results of Banded Systems of linear equations.

Table 4.15: Banded System n = 9

ITERATIVE METHODS n= 9				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	24	0.050000	1.1630e-004
Gauss-Seidel	0	13	0.050000	0.25000
SOR ω=1.12	0	8	0.015000	0.25000
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	5	0.021000	3.7993e-004
CES v with GS	0	9	0.017000	4.8422e-004
CES with SOR ω=1.11	0	9	0.041000	9.5845e-004
RES with Jacobi ω=0.9	0	26	0.021000	9.8621e-004
RES with GS ω=1.11	0	25	0.015000	7.6202e-005
RES with SOR w=1.11	0	28	0.015000	7.8968e-005
AGS with Jacobi	0	5	0.023000	2.5656e-016
AGS with Gauss-Seidel	0	5	0.016000	2.5656e-016
AGS with SOR ω= 0.9	0	5	0.015000	2.5656e-016
RS with Jacobi	0	2	0.031000	2.3024e-016
RS with Gauss-Seidel	0	2	0.509000	1.1766e-016
RS with SOR ω= 1.02	0	2	0.003000	8.8245e-017
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	1.2307e-016
Richardson Acceleration	0	5	0.047000	2.7e-016
Gmres	0	5	0.005000	2.6e-016
Minres	0	5	0.005000	4.5e-016
Qmr	0	5	0.005000	2.7e-016
Bicgstab	0	4.5	0.004000	1.4e-016

From Table 4.15,RES with Jacobi,RES with Gauss-Seideland RES with SOR did not improve the convergence of the stationary iterative methods.The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.12 respectively.

Table 4.16: Banded System n = 25

ITERATIVE METHODS n= 25				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	68	0.172000	3.0440e-005
Gauss-Seidel	0	38	0.140000	0.2500
SOR ω= 1.11	0	30	0.171000	0.2500
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	7	0.078000	8.3188e-004
CES with GS	1	100	0.858000	4.3157e+033
CES with SOR ω=1.25	1	100	0.946000	1.6346e+022
RES with Jacobi ω=1.25	1	101	0.063000	9.8092e+010
RES with GS ω=1.20	0	66	0.063000	2.4116e-005
RES with SOR ω=1.4	0	78	0.078000	3.1094e-005
AGS with Jacobi	0	10	0.015000	7.1249e-006
AGS with Gauss-Seidel	0	10	0.078000	7.1876e-016
AGS with SOR ω= 1.20	0	7.5	0.031000	5.9520e-007
RS with Jacobi	0	2	0.015000	1.8975e-006
RS with Gauss-Seidel	0	2	0.016000	1.5542e-006
RS with SOR ω=1.13	0	2	0.031000	1.3754e-006
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	5.0282e-016
Richardson Acceleration	0	3	0.047000	3.6478e-016
Gmres	0	10	0.015000	7.1e-006
Minres	0	10	0.061000	7.2e-016
Qmr	0	11	0.043000	5.8e-016
Bicgstab	0	7	0.014000	6e-007

From Table 4.16,CES with Gauss-Seidel, CES with SOR and RES with Jacobi did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 1.11 and 1.25 respectively.

Table 4.17 : Banded System n = 50

ITERATIVE METHODS n= 50				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	78	0.104580	3.1094e-005
Gauss-Seidel	0	63	0.202400	2.315e-004
SOR ω = 1.01	0	39	0.103000	2.453e-004
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	23	0.305000	6.0742e-004
CES with GS	0	12	0.203400	4.3194e-004
CES with SOR ω =0.9	0	10	0.204200	5.342e-004
RES with Jacobi ω =1.01	1	58	2.230000	6.0742e+235
RES with GS ω =0.9	1	45	1.303000	6.0742e+135
RES with SOR ω =1.11	0	16	0.230000	2.0132e-004
AGS with Jacobi	0	9	3.003300	3.0163e-004
AGS with Gauss-Seidel	0	8	1.020000	2.2052e-004
AGS with SOR ω = 0.9	0	7	0.116000	2.1452e-004
RS with Jacobi	0	2	0.230000	1.00000
RS with Gauss-Seidel	0	2	0.120300	1.00000
RS with SOR ω = 1.01	0	2	0.032200	1.00000
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	3.5e-007
Richardson Acceleration	0	3	0.015000	3.4e-007
Gmres	0	4	0.016000	9.7e-007
Minres	0	3	0.015000	9.7e-007
Qmr	0	4	0.015000	9.7e-007
Bicgstab	0	2.5	0.016000	1.4e-007

From Table 4.17,RES with Jacobi andRES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.11 respectively.

Table 4.18 : Banded System n = 100

ITERATIVE METHODS n= 100				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	89	3.014000	6.1424e-005
Gauss-Seidel	0	77	2.013000	4.0251e-005
SOR $\omega = 1.11$	0	62	1.003200	4.0432e-005
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	43	2.045000	3.0742e-005
CES with GS	0	36	1.023000	4.9232e-005
CES with SOR $\omega =0.9$	0	22	1.034000	2.4532e-005
RES with Jacobi $\omega =1.01$	1	57	5.300000	5.0722e+143
RES with GS $\omega =0.9$	1	48	6.020000	7.0342e+352
RES with SOR $\omega =1.25$	0	15	0.210000	6.0742e-005
AGS with Jacobi	0	9	0.023000	5.0432e-005
AGS with Gauss-Seidel	0	8	0.023200	3.1452e-005
AGS with SOR $\omega = 1.01$	0	7	0.012000	2.1612e-005
RS with Jacobi	0	2	0.013000	1.1375e-005
RS with Gauss-Seidel	0	2	0.012000	1.5542e-005
RS with SOR $\omega = 1.01$	0	2	0.031000	1.3754e-005
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.015000	3.3e-005
Richardson Acceleration	0	3	0.016000	4.7e-007
Gmres	0	5	0.015000	7.6e-008
Minres	0	4	0.015000	7.6e-008
Qmr	0	5	0.016000	7.6e-008
Bicgstab	0	2.5	0.015000	1.9e-007

From Table 4.18,RES with Jacobi,RES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.19 : Banded System n = 200

ITERATIVE METHODS n= 200				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	79	0.014000	4.1094e-005
Gauss-Seidel	0	63	0.014000	5.5143e-005
SOR ω = 1.01	0	59	0.015000	3.5265e-005
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	33	0.021000	7.0232e-004
CES with GS	0	25	0.015000	5.3434e-004
CES with SOR ω =1.11	0	24	0.032000	3.4324e-004
RES with Jacobi ω =0.9	1	59	1.644000	5.0742e+231
RES with GS ω =1.01	0	50	1.504400	6.0262e-004
RES with SOR ω =1.25	0	24	0.320000	2.0732e-004
AGS with Jacobi	0	17	0.246000	2.0132e-004
AGS with Gauss-Seidel	0	13	0.043000	5.1432e-004
AGS with SOR ω = 1.01	0	11	0.045100	2.4752e-004
RS with Jacobi	0	2	0.015000	1.8275e-006
RS with Gauss-Seidel	0	2	0.026000	1.5542e-006
RS with SOR ω = 0.9	0	2	0.031000	1.3454e-006
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	4.2e-007
Richardson Acceleration	0	3	0.015000	5.7e-007
Gmres	0	5	0.015000	1.5e-007
Minres	0	4	0.016000	1.5e-007
Qmr	0	5.5	0.015000	1.5e-007
Bicgstab	0	2.5	0.016000	3.3e-007

From Table 4.19, RES with Jacobi did not converge which corresponds with the largest values of the relative residual and the remaining methods improved the convergence of the stationary iterative methods.The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

Table 4.20: Banded System n = 500

ITERATIVE METHODS n= 500				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	98	0.020000	5.1043e-005
Gauss-Seidel	0	84	0.030000	4.434e-005
SOR ω = 1.11	0	70	0.035600	2.453e-005
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	33	0.025000	4.0322e-005
CES with GS	0	15	0.043000	5.5324e-005
CES with SOR ω =1.01	0	12	1.020000	5.4314e-005
RES with Jacobi ω =1.11	1	67	2.300000	6.0742e+234
RES n with GS ω =1.01	1	59	1.800000	5.0742e+124
RES with SOR ω =0.9	0	20	0.302000	6.0732e+231
AGS with Jacobi	0	15	0.023400	3.0322e-005
AGS with Gauss-Seidel	0	13	0.032000	2.1452e-005
AGS with SOR ω =	0	9	0.032000	3.1352e-005
RS with Jacobi	0	2	0.015000	1.3975e-006
RS with Gauss-Seidel	0	2	0.016000	1.5542e-006
RS with SOR ω = 0.9	0	2	0.031000	1.2654e-006
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.015000	2.7e-007
Richardson Acceleration	0	3	0.016000	5.7e-007
Gmres	0	5	0.016000	2.7e-007
Minres	0	4	0.031000	2.7e-007
Qmr	0	5	0.016000	2.7e-007
Bicgstab	0	2.5	0.015000	6e-007

From Table 4.20, RES with Jacobi,RES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.11 respectively.

Table 4.21 : Banded System n =800

ITERATIVE METHODS n= 800				
Methods	Flag	No. of Iteration	Speed(sec)	Relative Residual
Jacobi	0	88	0.053000	5.1365e-005
Gauss-Seidel	0	75	0.045000	3.543e-005
SOR ω = 1.25	0	44	0.020000	2.425e-005
ACCELERATION SCHEMES WITH ITERATIVE METHODS				
CES with Jacobi	0	23	0.046000	6.210e-005
CES with GS	0	16	0.032000	4.4932e-005
CES with SOR ω =1.01	0	11	0.012000	3.4353e-005
RES with Jacobi ω =1.25	1	68	2.450000	3.0742e+411
RES with GS ω =1.01	1	67	3.560000	5.0723e+401
RES with SOR ω =1.11	0	25	0.230000	4.0742e-004
AGS with Jacobi	0	12	0.430000	2.0142e-004
AGS with Gauss-Seidel	0	14	0.035000	2.1312e-004
AGS with SOR ω = 0.9	0	10	0.036000	2.4852e-005
RS with Jacobi	0	2	0.034000	1.8215e-005
RS with Gauss-Seidel	0	2	0.012000	1.5522e-005
RS with SOR ω = 1.01	0	2	0.021000	1.3324e-005
ACCELERATION METHODS				
Chebyshev Acceleration	0	2	0.016000	1.4536e-015
Richardson Acceleration	0	3	0.001500	2.7147e-016
Gmres	0	5	0.047000	3.7e-007
Minres	0	4	0.015000	3.7e-007
Qmr	0	5	0.047000	3.7e-007
Bicgstab	0	2.5	0.047000	7.4e-007

From Table 4.21, RES with Jacobi,RES with Gauss-Seidel did not converge which corresponds with the largest values of the relative residual.The remaining methods improved the convergence of the stationary iterative methods. The minimum and maximum optimal relaxation parameters for the stationary and acceleration schemes are 0.9 and 1.25 respectively.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

In this thesis, the performances of stationary iteration methods and accelerative schemes have been studied.Numerical experiments based on, Tridiagonal Banded and Dense SPD systems were carried out to verify the efficiency of acceleration schemes and the known Krylov subspace methods. From the numerical results, RS scheme and AGSimproved the convergence of the stationary iterative methods except RES and CES which did not improve the entire identified systems of linear equations. Finally, RS and AGShas the fastest convergence in terms of number of iteration. The findings for this research works are as follows:

1. The acceleration schemes were ranked in terms of efficiency of the algorithms: Residual smoothing scheme improved the convergence of the stationary iterative methods, followed by Accelerate Gradientscheme, Chebyshev Extrapolation scheme and finally Richardson extrapolation scheme.

2. The minimum and maximum optimal relaxation parameter for stationary and acceleration schemes are 0.9 and 1.25 respectively.

3. RES and CES failed to converge to with some of the chosen systems.

4. The schemes which fail to converged corresponded with large relative residual

5.2 Conclusions

The performance of some Acceleration schemesand stationary iterative methods have been assessedin terms of convergence, number of iteration, speed and relative residual which were applied to Tridiagonal systems, Dense SPD system and Banded system with varying dimensions. The maximum of iterations for the acceleration schemes (RS and AGS are 2 and 9 respectively.Again, number of iterations for the Krylov subspace methods: GMRES, QMR and BiCGSTAB wasless than or equal to the dimension of the coefficient matrix for each identified systems of linear equations. That is to say, if “k” represents the number of iteration and “n” the size of the coefficient matrix, then $k \leq n$. However, some of the acceleration schemes, especially CES and RES did not improve the convergence of some of the stationary iterative methods. These can be seen in the numerical resultsmore especially RES with Gauss-Seidel, RES with SOR, CES with Gauss-Seidel and CES with SOR. Again,Chebyshev acceleration and Richardson acceleration methods are the fastestconvergence methods in terms of number of iterations. Therefore, RS and AGSare very efficient in terms of number of iterations as compared to the known Krylov subspace methods when solving the identified linear systems.

5.3 Recommendations

It is recommended that;

1. Chebyshev and Richardson acceleration methods should be used when solving problems involving Banded, Tridiagonal and SPD systems.
2. RS, AGS with stationary method should be used when solving large and sparse systems of linear equations.

5.4Future work

Since RES did not improve the convergence of some of the identified systems, the further work is to:

1. Incorporate the Krylov subspace methods into RES to assess the convergence of stationary iterative methods.
2. Apply optimal relaxation parameter, ω with accelerated Overrelaxation scheme to speedup convergence of iterative methods.
3. Chebyshev and Richardson acceleration should also be applied to Hilbert systems

REFERENCES

Arioli, M., Demmel, J.W. and Duff, I.S. (1989). Solving sparse linear systems with sparse backward error. SIAM: Journal of Matrix Analysis and Applications. **10**(2):165–190

Auslender, A. andTeboulle, M. (2006) .Interior gradient and proximal methods for convex andconic Optimization SIAM J. Opt. 16(3):697–725.

Berman A, R. J. and Plemmons R.J, (1979).Nonnegative Matrices in theMathematical Sciences, Academic Press, New York, NY, USA,Burden, R.L. and Faires, J.D. (2011). Direct Methods for solving Systems. Numerical Methods4th Edition. Dublin City University.22-38.

Duan, L. and Zheng, B. (2008). Chebyshev polynomial acceleration of SOR solving the rank of Deficient linear system.. Department of Mathematics, Lanzhou UniversityLanzhou 730000, PR China. 1-16.

Fink, K.D. and Mathews, J.H. (2004).Numerical Methods Using Matlab 4th Edition.Person Education Inc, Upper Saddle River. New Jessey.

Gismalla, D.A. (2014). Matlab Software for Iterative Methods and Algorithms to Solve a Linear System. International Journal of Engineering and Technical Research (IJETR) 2(2): 2321-0869.

Gutknecht, M.H. and Rolin, S. (2002) .Chebyshev iterative Revisited. Parallel Computing. 28 (263 – 283).

Hadjidimos, A. (1978). Accelerated Overrelaxation Method. Mathematics of Computation. 32(141): 149-157.

Hageman, L.A.,and Young, D.M (1981). Applied iterative methods. Academic Press, New York.

Iqbal, J., Arif, M. and Ayaz, M.(2012). Preconditioned Iterative methods for Linear Systems.Life Science Journal 10(3):467-470

Jamil, N. (2012). A Comparison of Direct and Indirect Solvers for Linear Systems of equations International Journal of Emerging Science 2(2) 310-321.

Kalambi, I.B. (2008). A Comparison of Three Iterative Methods for the Solution of Linear System. Journal of Applied Sci, Environ. Manage. 12(4):53-55

Kelley, C.T. (1995). Iterative methods for linear and Nonlinear Equations.Society for Industrial and Applied Mathematics.

http://www4.ncsu.edu/~ctk/fr16_book.pdf

Kincaid, C. and Chenney, W. (2008).Numerical Mathematics and Computing. 6th Edition ISBN: 10-495-11475-8 pp. 280-293.

Li, W., Elsner, L.and Lu, L. (2002). Comparisons of spectral radiiand the theorem of Stein- Rosenberg. Linear Algebra and itsApplications, 348(3): 283–287

Peng, W. and Lin, Q. (2012). An acceleration method for stationary iterative solution of linearSystem of equation.Advances in Applied Mathematics and Mechanics 4(4):473-482.

Saad, Y. (1984). Acceleration Techniques for Solving Nonsymmetric Eigenvalue problem.Mathematics of Computing. 42(166): 567-588

Saad, Y. (2003). Iterative methods for sparse linear system. Second EditionSociety for Industrial and Applied Mathematics. ISBN 0899715342.

SalkuyehD.K (2007). Generalized Jacobi and Gauss-Seidel Methods for Solving LinearSystem of Equations.A Journal of Chinese Universities 16(2): 164-170

Santos, N.R. and Linhares, O.L. (1986).Convergence of Chebyshev Semi-Iterative methods.Journal of Applied Mathematics 16(59-68).North-Holland.

Tullius, T.K. (2011). Accelerated Discontinuous Garlerking Solvers with Chebyshev iterative Methods on the Graphical processing Unit.A Thesis submitted to Rice University

http://scholarship.rice.edu/bitstream/handle/1911/70478/TulliusT.pdf?sequence=1

Varga, R.S. (1962). Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, N.J

Walker, F.H. and Zhou, L. (1994).Residual Smoothing Techniques for Iterative Methods. SIAM Journal of Scientific Computing.15(2):297-312.

Wang, Z., Yang, C. and Yuan, Y. (2014).Convergence Results on Iteration Algorithms to Linear Systems. Scientific World Journal, Article ID 273873:10.

Wrigley, H.E. (1963). Accelerating the Jacobi method for solving simultaneous equations When the eigenvalue of iterative methods are complex. http://www.ams.org/mcom/1984-42-166/S0025-5718-1984-0736453-8

Yin,H. (2005). An iterative method for general variational inequalities.Journal of Industrial andManagement Optimization, 1(2): 201–209.

Zhang, H., Cui, Y. and Cu, J. (2004). An efficient Iterative Methods in Numerical CalculationJournal of Chemical and Pharmaceutical Research. 6(3): 179 – 187.

APPENDIX A Matlab Code for Jacobi, Gauss-Seidel and SOR

```
function x = Jacobi(A,b,x0,tol,max)
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x0=[0;0;0;0];
tol=0.001

max=100
tic
[n m] = size(A); xold = x0; C = -A;
for i=1:n
C(i,i) = 0;
end
for i=1:n
C(i,:) = C(i,:)/A(i,i);
end
for i=1:n
d(i,1) = b(i)/A(i,i);
end
i = 1;
while(i<=max)
xnew = C * xold + d;
if norm(xnew-xold) <= tol
        x = xnew;
disp('Jacobi method converged');
disp([i  xnew']);
return;
else
xold = xnew;
end
disp([i  xnew']);
    i=i+1;
    r=b-A*xold
    g=norm(r)/norm(b)
toc;
end
disp('Jacobi method did not converge');
disp('results after maximum number of iterations');
x = xnew;


function x = Seidel(A,b,x0,tol,max)
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x0=[0;0;0;0];
tol=0.001
w=1
max=100
[n m] = size(A);
x = x0;
tic
C = - A;
for i=1:n
C(i,i) = 0;
end
for i=1:n
C(i,:) = C(i,:)/A(i,i);
end
```

```
for i=1:n
r(i,1) = b(i)/A(i,i);
end
i = 1;
while(i<=max)
xold = x;
for j=1:n
x(j)=C(j,:) * x + r(j);
toc
end
if norm(xold-x) <=tol
disp('Gauss-Seidel method converged');
disp([i  x']);
return;
        r=b-A*x0
end
disp([i  x']);
    i=i+1;
    g=norm(r)/norm(b)
end
disp('Gauss-Seidel method did not converge');



function x = SOR(A,b,x0,w,tol,max)
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x0=[0;0;0;0];
tol=0.001
w=1.11
max=100
[n m] = size(A);
x = x0;
tic
C = -A;
for i=1:n
C(i,i) = 0;
    r=b-A*x0
end
for i=1:n
C(i,1:n)=C(i,1:n)/A(i,i);
end
for i=1:n
r(i,1) =b(i)/A(i,i);
end
i=1;
while (i<=max )
xold=x;
for j=1:n
        x(j)=(1-w)*xold(j)+w*(C(j,:)*x+r(j));
end
if norm(xold-x)<=tol
disp('SOR method converged');
disp([i   x']);
return;
end
disp([i   x']);
    i=i+1;
    g=norm(r)/norm(b)
toc;
end
disp('SOR method did not converged');
```

APPENDIX B Matlab Code for Acceleration Methods

Chebyshev Acceleration

```
function [x, error, iter, flag] = cheby(A, x, b, M, max_it, tol)
tic
A=matrix(A)
b=[RHS]
x=[intial vector]
max_it=100
tol=0.001
alpha=0
iter = 0                                % initialization
flag = 0;
  bnrm2 = norm( b );
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end
  r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

eigs = eig( inv(A)*A );
eigmax = max( eigs );
eigmin = min( eigs );

  c = ( eigmax - eigmin ) / 2.0;
  d = ( eigmax + eigmin ) / 2.0;

for iter = 1:max_it,                     % begin iteration

    z =  A \ r;

if ( iter > 1 )                        % direction vectors
beta = ( c*alpha / 2.0 )^2;
alpha = 1.0 / ( d - beta );
       p = z + beta*p;
else
       p = z;
alpha = 2.0 / d
end

x   = x + alpha*p                    % update approximation
n_iteration=iter
    r = r - alpha*A*p
error = norm(r) / bnrm2            % check convergence
if ( error <= tol  ), break, end
toc;
end

if ( error > tol ) flag = 1

end;        % no convergence
```

Richardson acceleration

```
function [x,noit,err,Q,rho] = Richardson(A,b,x0,eps,maxit);
% [x,noit,err] = Richardson(A,b,x0,eps,maxit);
%
% Purpose: Solve Ax=b using Richardson iteration, with initial guess x0
%           to tolerance eps in less that maxit iterations.
%
% Output : x    Solution
%          noit Number of iterations
%          err  L2 of difference between last two iterations
%
 A=matrix(A)
  b=RHS
x0=initial vector
eps=0.001
maxit=1000
w=1.25
M=diag(diag(A))
N=M-A
T=inv(M)*N
C=inv(M)*b
% Create splitting matrix
Q = eye(n);
% Compute reduction factor
R = inv(Q)*(Q-A);
rho = max(abs(eig(R)))
tic
% Set initial values
x= x0;
noit = 1;
err = 1.0;
while (err>eps)
xn = (((1-w)*Q+w*T)*x) + w*C
err = norm(xn-x)
noit = noit+1
    x = xn
    r=b-A*x
    Relres=norm(r)/norm(b)
if (noit>maxit) break; end;
   x=qmr(A,b)
toc;
end;
return
```

APPENDIX C Matlab Code for Acceleration Schemes

Chebyshev with Jacobi

```
function [x, error, iter, flag] = cheby(A, x, b, M, max_it, tol)
 A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x=[0;0;0;0]
max_it=100
tic
R=chol(A)
R'
M=R*R'
tol=0.001
alpha=0
iter = 0                                        % initialization
flag = 0;
  bnrm2 = norm( b );
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end
  r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

eigs = eig( inv(M)*A );
eigmax = max( eigs );
eigmin = min( eigs );

  c = ( eigmax - eigmin ) / 2.0;
  d = ( eigmax + eigmin ) / 2.0;

for iter = 1:max_it,                             % begin iteration

    z =  M \ r;

if ( iter > 1 )                                % direction vectors
beta = ( c*alpha / 2.0 )^2;
alpha = 1.0 / ( d - beta );
       p = z + beta*p;
else
       p = z;
alpha = 2.0 / d
end

x   = x + alpha*p                            % update approximation
n_iteration=iter
    r = r - alpha*A*p
error = norm(r) / bnrm2                    % check convergence
if ( error <= tol  ), break, end
toc;
end

if ( error > tol ) flag = 1
```

```
end;        % no convergence
Cheby with Gauss-Seidel

function [x, error, iter, flag] = cheby(A, x, b, M, max_it, tol)
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x=[0;0;0;0]
max_it=100
tic
R=chol(A)
R'
M1=R*R'
L1=[0 0 0 0;-3 0 0 0;1 -1 0 0;-2 4 1 0]
L=inv(M1)*L1
M=M1+L
tol=0.001
alpha=0
iter = 0                                % initialization
flag = 0;
  bnrm2 = norm( b );
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end
  r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

eigs = eig( inv(M)*A );
eigmax = max( eigs );
eigmin = min( eigs );

  c = ( eigmax - eigmin ) / 2.0;
  d = ( eigmax + eigmin ) / 2.0;

for iter = 1:max_it,                     % begin iteration

    z =  M \ r;

if ( iter > 1 )                        % direction vectors
beta = ( c*alpha / 2.0 )^2;
alpha = 1.0 / ( d - beta );
       p = z + beta*p;
else
       p = z;
alpha = 2.0 / d
end

x  = x + alpha*p                     % update approximation
n_iteration=iter
    r = r - alpha*A*p
error = norm(r) / bnrm2            % check convergence
if ( error <= tol  ), break, end
toc;
end

if ( error > tol ) flag = 1
```

```
end;         % no convergence
```

Cheby with SOR

```
function [x, error, iter, flag] = cheby(A, x, b, M, max_it, tol)
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x=[0;0;0;0]
max_it=100
w=1.22
tic
R=chol(A)
R'
M1=R*R'
L1=[0 0 0 0;-3 0 0 0;1 -1 0 0;-2 4 1 0]
L=inv(M1)*L1
M2=M1+L
M=w*M2
tol=0.001
alpha=0
iter = 0                                % initialization
flag = 0;
  bnrm2 = norm( b );
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end
  r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

eigs = eig( inv(M)*A );
eigmax = max( eigs );
eigmin = min( eigs );

  c = ( eigmax - eigmin ) / 2.0;
  d = ( eigmax + eigmin ) / 2.0;

for iter = 1:max_it,                     % begin iteration

    z =  M \ r;

if ( iter > 1 )                        % direction vectors
beta = ( c*alpha / 2.0 )^2;
alpha = 1.0 / ( d - beta );
       p = z + beta*p;
else
       p = z;
alpha = 2.0 / d
end

x  = x + alpha*p                     % update approximation
n_iteration=iter
    r = r - alpha*A*p
error = norm(r) / bnrm2            % check convergence
if ( error <= tol  ), break, end
toc;
```

```
end

if ( error > tol ) flag = 1
end;         % no convergence
```

Richardson with Jacobi

```
function [x,noit,err,Q,rho] = Richardson(A,b,x0,eps,maxit);
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x0=[0;0;0;0];
eps=0.001
maxit=100
w=1.12
M=diag(diag(A))
N=M-A
T=inv(M)*N
C=inv(M)*b
% Create splitting matrix
Q = eye(4);
% Compute reduction factor
R = inv(Q)*(Q-A);
rho = max(abs(eig(R)))
% Set initial values
x= x0;
noit = 1;
err = 1.0;
while (err>eps)
xn = (((1-w)*Q+w*T)*x) + w*C
err = norm(xn-x)
noit = noit+1
    x = xn
tic
    r=b-A*x
    Relres=norm(r)/norm(b)
if (noit>maxit) break; end;
toc;
end;
return
```

Richardson with GS

```
function [x,noit,err,Q,rho] = Richardson(A,b,x0,eps,maxit);
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x0=[0;0;0;0];
eps=0.001
maxit=100
w=1.01
M1=diag(diag(A))
L1=[0 0 0 0;-3 0 0 0;1 -1 0 0;-2 4 1 0]
L=inv(M1)*L1
M=M1+L
N=M-A
T=inv(M)*N
```

```
C=inv(M)*b
% Create splitting matrix
Q = eye(4);
% Compute reduction factor
R = inv(Q)*(Q-A);
rho = max(abs(eig(R)))
% Set initial values
x= x0;
noit = 1;
err = 1.0;
tic
while (err>eps)
xn = (((1-w)*Q+w*T)*x) + w*C
err = norm(xn-x)
noit = noit+1
    x = xn
    r=b-A*x
    Relres=norm(r)/norm(b)
if (noit>maxit) break; end;
toc;
end;
return
```

Richardson with SOR

```
function [x,noit,err,Q,rho] = Richardson(A,b,x0,eps,maxit);
A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4];b=[1;1;1;1];
x0=[0;0;0;0];
eps=0.001
maxit=100
w=1.25
M1=diag(diag(A))
L1=[0 0 0 0;-3 0 0 0;1 -1 0 0;-2 4 1 0]
L=inv(M1)*L1
M2=M1+L
M=w*M2
N=M-A
T=inv(M)*N
C=inv(M)*b
% Create splitting matrix
Q = eye(4);
% Compute reduction factor
R = inv(Q)*(Q-A);
rho = max(abs(eig(R)))
% Set initial values
x= x0;
noit = 1;
err = 1.0;
tic
while (err>eps)
xn = (((1-w)*Q+w*T)*x) + w*C
err = norm(xn-x)
noit = noit+1
    x = xn
    r=b-A*x
    Relres=norm(r)/norm(b)
if (noit>maxit) break; end;
toc;;
end;
return
```

APPENDIX D Banded System Code

```
n=(800)
rand('state',0);
A1=zeros(n,n);
for i=1:n;
for j=1:n;if i==j;A1(i,j)=40;elseif i>j A1(i,j)=1+rand; else
A1(i,j)=0; end
end
end
A=A1+(tril(A1,-1))'
b=(A-eye(n))*ones(n,1);
```

APPENDIX E Dense and SPD System Code

```
n=(9);
rand('state',0);
R=rand(n, n);
b=rand(n, 1);
A=R'*R+n*eye(n);
```

```
n=(25)

rand('state',0);

A1=zeros(n,n);

for i=1:n;

for j=1:n;if i==j;A1(i,j)=40;elseif i>j A1(i,j)=1+rand; else

A1(i,j)=0; end

end

end

A=A1+(tril(A1,-1))'

b=(A-eye(n))*ones(n,1);
```

APPENDIX F Tridiagonal System Code

n=25

```
for n = [25]
    a = -ones(n-1,1);
    b = 5*ones(n,1);

    c = a;
% Create a tridiagonal matrix with 2's on the diagonal and -1's on the
% sub- and superdiagonal.
    A1 = gallery('tridiag',a,b,c)
    A=full(A1)
% A random vector for the right-hand side
    d = rand(n,1)

end
```

n=9

A=[-4,1,0,1,0,0,0,0,0;1,-4,1,0,1,0,0,0,0;0,1,-4,0,0,1,0,0,0;1,0,0,-4,1,0,1,0,0;0,1,0,1,-4,1,0,1,0;0,0,1,0,1,-4,0,0,1;0,0,0,1,0,0,-4,1,0;0,0,0,0,1,0,1,-4,0;0,0,0,0,0,1,0,1,-4]

b=[-100;-100;-100;0;0;0;0;0;0]

APPENDIX G Computer Specification

Processor : Intel(R) Pentium (R) CPU P600 @ 1.87GHz 1.87

Installed Memory (RAM) : 4.00GB (2.99GB Usable)

System Type : 32-bit Operating System.